CS 228 – Biometric Security with AI

Assignment 2

Due: Dec 4th, 2025

There is a single task in this assignment. Scope of this assignment is reduced to provide time for project work.

Task Description – (100 points)

Extension of data poisoning, to a multiclass version (in class we did a two class version). Perform a clean-label poisoning attack on CIFAR-10 dataset. For this task, you can use the shared code on canvas as a starter or generate code using AI assistants (but make sure to understand it before using and modifying it). The final code must be fully understood and approved by you, and you should be able to explain it to me, if asked. You can also use the google education credits for this assignment is your local hardware is not suitable.

1. **Data Preparation:** Select a subset of CIFAR-10 with 4 classes (use only a small set of images from each class, approx. 200 to 500 depending on your system capacity). Remove 10 *target images* (of class T) from the training set and set it aside. Pick 10 *base images* b from a different class B. All labels of poisons must remain the original (base) class label.
2. **Model Architecture:** Build a small CNN (e.g. 2–3 conv layers + 1–2 FC layers) suitable for CIFAR-10. The network should output logits for all chosen classes.
3. **(20 points) Initial Training:** Train the CNN on the (clean) training subset (excluding the target). Verify that it correctly classifies a held-out validation set and the target image (before poisoning). Record baseline accuracy and confirm the targets are initially classified as its true class.
4. **(20 points) Poison Generation:** Implement the iterative poisoning algorithm for each poison (as in Poison Frogs paper). In each iteration:
    - **Forward step:** Compute features f(x) of the current poison x and of target t using the model, then take a gradient descent step on x to minimize |f(x)-f(t)|^2.
    - **Backward step:** Adjust x to stay close to the base image b (e.g. via forbenius norm used in paper, also in example code).
    - Optionally clamp or clip x to valid image range.

Continue this for a set number of iterations (e.g. 100–200) or until |f(x)-f(t)| are very small. **Visualize** the poison images at several iterations (e.g. every 20 iterations) using matplotlib to show changes.

5. **(20 points) Retraining and Evaluation:** Insert the final poisons into the training set (with label B) and retrain the CNN from scratch (or fine-tune). Evaluate on the test set and on the target image. The attack is **successful** if the target t is now misclassified as class B (the base class) by the model. Compare the model's overall accuracy (should remain high) and report the stats on individual classes.

6. **(40 points) Report Results:** Provide plots of the poisons evolving, the final poison images, and relevant metrics (e.g. accuracy before/after, target prediction). Discuss whether some targets were misclassified and why.

Submit code and report.

The report can be brief 3 page, mainly with graphs, tables (performance metrics) and poison images.